

# FSA-CRP Data Generation

As Built Documentation

Soil Survey Division  
November 2002



<b>INTRODUCTION.....</b>	<b>5</b>
<b>A NOTE ABOUT REPRESENTATIVE VALUES .....</b>	<b>6</b>
<b>DATA SELECTION AND OUTPUT.....</b>	<b>7</b>
<b>DERIVATION OF EACH ATTRIBUTE.....</b>	<b>9</b>
STCTY .....	9
SSAID .....	9
MUSYM.....	9
MU_LLEAF.....	9
MU_LEACH.....	10
MU_IFACTOR.....	12
MU_KFACTOR.....	12
MU_LS.....	12
MU_TFACTOR.....	16
SOURCE .....	16
<b>TEST PLAN .....</b>	<b>17</b>
MAP UNIT SELECTION TESTS .....	17
CORRESPONDING COUNTY RELATED TESTS .....	18
COMPONENT SELECTION AND T FACTOR TESTING.....	19
LONG LEAF PINE SUITABILITY TESTING .....	20
I FACTOR TESTING.....	21
K FACTOR TESTING .....	21
LS FACTOR TESTING.....	22
UNDRAINED LEACHING INDEX TESTING .....	23
<b>THE GORY DETAILS.....</b>	<b>24</b>
DECLARATIONS.....	24
CASESENSITIVEMUSYM.....	24
FSACRP_CREATE_DATA .....	25
INITLSTABLE.....	43
INPALOUSE .....	52
ISLONGLEAFPINECOUNTY .....	53
RV .....	53



## Introduction

This document was prepared by Dorn Egley, technical analyst at the NRCS Information Technology Center in Fort Collins, Colorado, at the request of the Soil Survey Division. It explains the derivation rules used to create the soils data export delivered to the Farm Services Agency for use in the 2003 Conservation Reserve Program. In addition to an explanation of each individual derivation, a copy of the actual code is included as an ultimate “ground truth” of the derivation logic.

Below is the format of the table/file that is generated from the MS Access SSURGO template database.

Column Position	Column Label	Definition
1	stcty	Concatenated state code (alpha FIPS) and county code (numeric FIPS) (variable character, maximum length 5). Not NULL.
2	ssaid	Soil survey area ID (variable character, maximum length 5). Not NULL
3	musym	Mapunit symbol (variable character, maximum length 6). Not NULL.
4	mu_lleaf	Partial criteria for long leaf pine suitability (character, maximum length 1). Domain Y, N. Not NULL.
5	mu_leach	Partial N2b criteria – undrained leaching index (integer, maximum length 1). Domain is 1, 2, 3 or null.
6	mu_ifactor	Partial N5a criteria – wind erodibility index (integer, maximum length 3). Domain 0, 38, 48, 56, 86, 134, 160, 180, 220, 250, 310 or null.
7	mu_kfactor	Erodibility factor (floating point, maximum length 1 plus two decimal places). Domain 0.02, 0.05, 0.10, 0.15, 0.17, 0.20, 0.24, 0.28, 0.32, 0.37, 0.43, 0.49, 0.55, 0.64 or null.
8	mu_LS	Topographic factor determined from Agriculture Handbook 703, Table 4-2 (floating point, maximum length 2 plus two decimal places). Domain restricted to values in Table 4-2 or null.
9	mu_tfactor	Soil loss tolerance factor (integer, maximum length 1). Domain 1 – 5 or null.
10	source	Data source flag (character, maximum length 1). Domain (N, S) for records created directly from NASIS or from NASIS export in SSURGO v2 Format. Not NULL

## A Note About Representative Values

Many derivations are based in part on one or more derived representative values, or what we refer to as “RV value”, i.e. “RV hzdept”, “RV om”, etc. In every case, the algorithm for deriving the RV value is the same:

```
If value_r is not null then
    return value_r
Else if value_l is not null and value_h is not null then
    return (value_l + value_h)/2
Else
    return Null
End If
```

Whenever a derived RV value is part of the criteria for selecting records, records where the derived RV value is null are always excluded. This is why it is possible that we cannot determine which component to use for a particular map unit.

## Data Selection and Output

The FSA-CRP data generation functions always process ALL data that currently resides in the MS Access SSURGO template database. In other words, there is no capability to generate FSA-CRP data for some subset of the data in a template database.

Each output record represents a map unit that occurs in a particular county. The attribute values correspond to the dominant component of that map unit, based on RV percent composition. If there is a tie for dominant component, the component with the lowest key value (cokey) is arbitrarily selected.

If the survey contains more than one map unit with the same map unit symbol, which map unit is selected is based on map unit status, with preference given as follows:

Correlated > Approved > Provisional

Additional symbols are excluded from consideration.

The process of determining what county or counties a map unit occurs in is a multi-part process. If a map unit in a survey area has at least one county record in the Map Unit Area Overlap table (muaoverlap), a record is output for each county corresponding to that map unit.

If a map unit has no county record in the Map Unit Area Overlap table, but the corresponding survey area has at least one county record in the Legend Area Overlap table (laoverlap), a record is output for that map unit for each county corresponding to that survey area.

If a survey area has no corresponding county record in the Legend Area Overlap table, a record is output for that map unit for each record that the corresponding survey area has in the Survey Area-County Geographic Coincidence table (SYSTEM – Survey Area-County Geographic Coincidence).

If a map unit has no county record in the Map Unit Area Overlap table, and the corresponding survey area has no county record in the Legend Area Overlap table, and the corresponding survey area has no record in the Survey Area-County Geographic Coincidence table, NO record for that map unit is output. If this occurs for any map unit, a warning dialog is displayed at the end of the data generation process. Map units for which no corresponding county could be determined are logged in the file named “SYSTEM – FSA-CRP – Warnings and Rejects”.

It is also possible that no dominant component for a map unit can be determined. If this occurs for any map unit, a warning dialog is displayed at the end of the data generation process. Map units for which no dominant component can be determined are also logged to the table named “SYSTEM – FSA-CRP – Warnings and Rejects”. As long as a

corresponding county can be determined, such a map unit is output, but the values of all FSA-CRP criteria attributes will be null.



## Derivation of Each Attribute

### ***stcty***

If this map unit has at least one corresponding county record in table “muaoverlap”, then the value of “stcty” is set to “laoverlap.areasymbol” for each corresponding map unit area overlap record. In other words, if this map unit has more than one county record in table “muaoverlap”, more than one record for this map unit will be output.

If this map unit has no corresponding county record in table “muaoverlap”, then the value of “stcty” is set to “SYSTEM - Survey Area-County Geographic Coincidence.stcoid” for each record that the corresponding survey area has in table “SYSTEM - Survey Area-County Geographic Coincidence”. In other words, if the corresponding survey area has more than one record in table “SYSTEM - Survey Area-County Geographic Coincidence”, more than one record for this map unit will be output.

If the map unit has no corresponding county record in table “muaoverlap”, and the corresponding survey area has no corresponding record in table “SYSTEM - Survey Area-County Geographic Coincidence”, no record for this map unit will be output. Map units for which no corresponding county can be determined are logged to table “SYSTEM – FSA-CRP – Warnings and Rejects”, and the user is notified.

### ***ssaid***

The value of “ssaid” is always set to the value of “legend.areasymbol” of the legend record corresponding to this map unit.

### ***musym***

The value of “musym” is always set to the value of “mapunit.musym” of the corresponding map unit record.

### ***mu\_lleaf***

All fields used in this derivation are from the selected component of the corresponding map unit.

If longleaf pine is explicitly referenced (“table.plantsym” = “PIPA2”) in any of the following tables for this component, return “Y”:

Component Trees to Manage (cotreestomng)  
Component Existing Plants (coeplants)  
Component Forest Productivity (coforprod)

Else if all of the following are true, return “Y”:

There is no soil moisture layer where RV moisture layer depth to top is < 30 cm  
and “cosoilmoist.soismoiststat” = “Wet”.

There is at least one soil layer where RV horizon depth to top is < 30 that has one  
of the following soil textures:

- coarse sand
- sand
- fine sand
- very fine sand
- loamy coarse sand
- loamy sand
- loamy fine sand
- loamy very fine sand
- coarse sandy loam
- sandy loam
- fine sandy loam
- very fine sandy loam

There is no soil layer where RV horizon depth to top is < 30 cm and (RV pH 1:1  
H2O >= 6 or RV pH 1:1 H2O is null).

Else return “N”.

In a final step, upon output, suitability for long leaf pine is automatically reset to “N”  
unless the corresponding county occurs in the list of counties contained in table  
“SYSTEM – Long Leaf Pine Counties”.

### ***mu\_leach***

All fields used in this derivation are from the selected component of the corresponding  
map unit.

If “component.taxorder” = “Histosols”, return 1.

Else if “component.hydgrp” is not null, and “component.hydgrp” is in (“D”, “A/D”,  
“B/D”, “C/D”), return 1.

Else if “component.hydgrp” is not null, and derived “kval” (see below) is not null, and derived “calc” (see below) is not null, return Undrained Leaching Index based on the following formula:

```
If (hg = "D" Or hg = "A/D" Or hg = "B/D" Or hg = "C/D") Or
(hg = "C" And calc <= 10000 And kval >= 280) Or
(hg = "C" And calc >= 10000) Or
(hg = "B" And calc >= 35000 And kval >= 400) Or
(hg = "B" And calc >= 45000 And kval >= 200) Then
    mu_leach = 1
ElseIf (hg = "A" And calc <= 30000) Or
(hg = "B" And calc <= 9000 And kval <= 480) Or
(hg = "B" And calc <= 15000 And kval <= 260) Then
    mu_leach = 3
ElseIf (hg = "A" And calc > 30000) Or
(hg = "B" Or hg = "C") Then
    mu_leach = 2
End If
```

Else return null.

OK, the first line of the formula appears to be redundant with the second if clause. Sue me.

The attribute “calc” is defined as:

$$\text{RV organic matter} * (\text{RV horizon depth to bottom} / 2.54) * 1000$$

The attribute “kval” is defined as:

$$\text{“chorizon.kwfact”} * 1000$$

RV organic matter, RV horizon depth to bottom and “chorizon.kwfact” are all from the first layer where “chorizon.kwfact” is not null, “chorizon.desgnmaster” <> “O” and RV organic matter <= 35, and layers are sorted by RV horizon depth to top ascending, “chorizon.kwfact” descending. If no layer meets this criteria, RV organic matter, RV horizon depth to bottom and “chorizon.kwfact” are all null, and therefore “calc” and “kval” are null.

***mu\_ifactor***

If “component.wei” for the selected component of the corresponding map unit is not null, return “component.wei”.

Else if “component.wei” for the selected component of the corresponding map unit is null, but “component.weg” is not null, return the value of I Factor based on the following lookup table.

WEG	I Factor
1	220
2	134
3	86
4	86
4L	86
5	56
6	48
7	38
8	0

Else if “component.wei” and “component.weg” for the selected component of the corresponding map unit are both null, return null.

***mu\_kfactor***

The intent for K Factor is to return K Factor for the first mineral layer of a soil that is not a Histosol. All fields used in this derivation are from the selected component of the corresponding map unit.

If “component.taxorder” = “Histosols”, return null.

Else select the first layer where either “chorizon.kffact” or “chorizon.kwfact” is not null, “chorizon.designmaster” <> “O” and RV organic matter <= 35, and layers are sorted by RV horizon depth to top ascending, “chorizon.kffact” descending and “chorizon.kwfact” descending. If “chorizon.kffact” is not null, return “chorizon.kffact”, else return “chorizon.kwfact”.

If no layer meets the selection criteria, return null.

***mu\_LS***

All fields used in this derivation are from the selected component of the corresponding map unit.

If RV slope is null, return null.

Else establish RV slope length and lookup LS Factor using RV slope and RV slope length using the following table:

## FSA-CRP Data Generation As Built Documentation

Slope Range	Slope Length in feet																
	>= 0 & < 4.5	>= 4.5 & < 7.5	>= 7.5 & < 10.5	>= 10.5 & < 13.5	>= 13.5 & < 20	>= 20 & < 37.5	>= 37.5 & < 62.5	>= 62.6 & < 87.5	>= 87.5 & < 125	>= 125 & < 175	>= 175 & < 225	>= 225 & < 275	>= 275 & < 350	>= 350 & < 500	>= 500 & < 700	>= 700 & < 900	>= 900
>= 0 & < 0.35	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.05	00.06	00.06	00.06
>= 0.35 & < 0.75	00.07	00.07	00.07	00.07	00.07	00.08	00.08	00.08	00.09	00.09	00.09	00.09	00.09	00.10	00.10	00.10	00.10
>= 0.75 & < 1.5	00.11	00.11	00.11	00.11	00.11	00.12	00.13	00.14	00.14	00.15	00.16	00.17	00.17	00.18	00.19	00.20	00.20
>= 1.5 & < 2.5	00.17	00.17	00.17	00.17	00.17	00.19	00.22	00.25	00.27	00.29	00.31	00.33	00.35	00.37	00.41	00.44	00.47
>= 2.5 & < 3.5	00.22	00.22	00.22	00.22	00.22	00.25	00.32	00.36	00.39	00.44	00.48	00.52	00.55	00.60	00.68	00.75	00.80
>= 3.5 & < 4.5	00.26	00.26	00.26	00.26	00.26	00.31	00.40	00.47	00.52	00.60	00.67	00.72	00.77	00.86	00.99	01.10	01.19
>= 4.5 & < 5.5	00.30	00.30	00.30	00.30	00.30	00.37	00.49	00.58	00.65	00.76	00.85	00.93	01.01	01.13	01.33	01.49	01.63
>= 5.5 & < 7	00.34	00.34	00.34	00.34	00.34	00.43	00.58	00.69	00.78	00.93	01.05	01.16	01.25	01.42	01.69	01.91	02.11
>= 7 & < 9	00.42	00.42	00.42	00.42	00.42	00.53	00.74	00.91	01.04	01.26	01.45	01.62	01.77	02.03	02.47	02.83	03.15
>= 9 & < 11	00.46	00.48	00.50	00.51	00.52	00.67	00.97	01.19	01.38	01.71	01.98	02.22	02.44	02.84	03.50	04.06	04.56
>= 11 & < 13	00.47	00.53	00.58	00.61	00.64	00.84	01.23	01.53	01.79	02.23	02.61	02.95	03.26	03.81	04.75	05.56	06.28
>= 13 & < 15	00.48	00.58	00.65	00.70	00.75	01.00	01.48	01.86	02.19	02.76	03.25	03.69	04.09	04.82	06.07	07.15	08.11
>= 15 & < 18	00.49	00.63	00.72	00.79	00.85	01.15	01.73	02.20	02.60	03.30	03.90	04.45	04.95	05.86	07.43	08.79	10.02
>= 18 & < 22.5	00.52	00.71	00.85	00.96	01.06	01.45	02.22	02.85	03.40	04.36	05.21	05.97	06.68	07.97	10.23	12.20	13.99
>= 22.5 & < 27.5	00.56	00.80	01.00	01.16	01.30	01.81	02.82	03.65	04.39	05.69	06.83	07.88	08.86	10.65	13.80	16.58	19.13
>= 27.5 & < 35	00.59	00.89	01.13	01.34	01.53	02.15	03.39	04.42	05.34	06.98	08.43	09.76	11.01	13.30	17.37	20.99	24.31
>= 35 & < 45	00.65	01.05	01.38	01.68	01.95	02.77	04.45	05.87	07.14	09.43	11.47	13.37	15.14	18.43	24.32	29.60	34.48
>= 45 & < 55	00.71	01.18	01.59	01.97	02.32	03.32	05.40	07.17	08.78	11.66	14.26	16.67	18.94	23.17	30.78	37.65	44.02
>= 55	00.76	01.30	01.78	02.23	02.65	03.81	06.24	08.33	10.23	13.65	16.76	19.64	22.36	27.45	36.63	44.96	52.70

The above table was derived from table 4-2 in Agricultural Handbook 703, “Values for topographic factor, LS, for moderate ratio of rill to interrill erosion”.

If the derived RV value of “component.sloplenuse” is not null, use that value for RV slope length.

Else if the corresponding survey area IS NOT in the Palouse region, use RV slope to lookup RV slope length using the following table:

<b>Slope Range</b>	<b>RV Slope Length in feet</b>
rvslope >= 0 And rvslope < 0.75	100
rvslope >= 0.75 And rvslope < 1.5	200
rvslope >= 1.5 And rvslope < 2.5	300
rvslope >= 2.5 And rvslope < 3.5	200
rvslope >= 3.5 And rvslope < 4.5	180
rvslope >= 4.5 And rvslope < 5.5	160
rvslope >= 5.5 And rvslope < 6.5	150
rvslope >= 6.5 And rvslope < 7.5	140
rvslope >= 7.5 And rvslope < 8.5	130
rvslope >= 8.5 And rvslope < 9.5	125
rvslope >= 9.5 And rvslope < 10.5	120
rvslope >= 10.5 And rvslope < 11.5	110
rvslope >= 11.5 And rvslope < 12.5	100
rvslope >= 12.5 And rvslope < 13.5	90
rvslope >= 13.5 And rvslope < 14.5	80
rvslope >= 14.5 And rvslope < 15.5	70
rvslope >= 15.5 And rvslope < 17.5	60
rvslope >= 17.5	50

The above table was derived from a table that was originally provided by Lightle and Weesies, 10/1/1996.

Else if the corresponding survey area IS in the Palouse region, use RV slope to lookup RV slope length using the following table:

The following slope lengths for the “Palouse” (MLRA 9) area were determined by Tom Gohlke in consultation with Don McCool, ARS and Harry Riehle. Tom says, “Keep in mind that many real LS’s in the field are complex slopes and consist of combinations of these slopes. For instance, it is common to find an “L” beginning on a 2%-5% slope and extending onto and ending on a 21%-25% slope. The total “L” may be less than the sum of the values for these two segments as shown in the following table.”

<b>Slope Range</b>	<b>RV Slope Length in feet</b>
rvslope >= 0 And rvslope < 5.5	350
rvslope >= 5.5 And rvslope < 10.5	275
rvslope >= 10.5 And rvslope < 15.5	225
rvslope >= 15.5 And rvslope < 20.5	175
rvslope >= 20.5 And rvslope < 25.5	150
rvslope >= 25.5 And rvslope < 35.5	125
rvslope >=35.5	100

### ***mu\_tfactor***

The value of “mu\_tfactor” is always set to “component.tfact” of the selected component of the corresponding map unit record.

### ***source***

Source is used to distinguish the source of a particular FSA-CRP data record. Russ Kelsea wanted to be able to distinguish between data produced from NASIS (source = “N”) and data produced from a SSURGO template database (source = “S”).

The value of “source” is always set to “S”.



## Test Plan

My FSA-CRP test data includes three legends, in the transactional NASIS database, for the following three geographic areas:

Area Type NASIS Site	Area Type Name	Area Symbol	Area Name
NSSC Data	Dorn's FSA-CRP Test SSA Type	XX001	Shire, Middle Earth
NSSC Data	Dorn's FSA-CRP Test SSA Type	XX002	Mordor, Middle Earth
NSSC Data	Dorn's FSA-CRP Test SSA Type	XX003	Rohan, Middle Earth

Survey area XX001 contains only one map unit whose symbol is 1. This map unit has no corresponding records in the map unit area overlap table, and this survey area has no corresponding records in the survey area-county geographic coincidence table. The entire purpose of this survey area is to demonstrate that no output record for a map unit is produced when no corresponding county can be determined. In the Access database, such a map unit is logged to table "SYSTEM – FSA-CRP – Warnings and Rejects".

Survey area XX002 contains the bulk of the test data.

Survey XX003 is a survey in the Palouse region. The only map units in this survey are used for testing LS Factor in the Palouse region.

I went with "XX" as the state portion of the area symbol so that these area symbols won't conflict with any real area symbol, as far as adding test records to the survey area-county geographic coincidence table and the Palouse region SSA table.

### ***Map Unit Selection Tests***

Map units with non-numeric symbols are used to check that the correct map unit is selected in cases where there is more than one map unit with the same symbol. This set of map units also verifies that a map unit with status "additional" is not output. Because the output file does not contain mustatus, all non-numeric map units that should be selected have a corresponding T Factor of 1, and all non-numeric map units that should not be selected have a corresponding T Factor of 5.

This set of map units also tests that case sensitivity is being taken into account in the Access database. This is not a concern in NASIS, since NASIS is case sensitive by default.

<b>Musym</b>	<b>Mustatus</b>	<b>T factor</b>	<b>Output?</b>
HOA	Approved	5	No
HOA	Correlated	1	Yes
HoA	Provisional	5	No
HoA	Correlated	1	Yes
hoA	Provisional	5	No
hoA	Approved	1	Yes
hoa	Additional	5	No

### ***Corresponding County Related Tests***

A map unit should not be output unless a corresponding county or counties can be determined.

Survey area XX001 contains one map unit whose symbol is “1”. This map unit has no corresponding county record in the map unit area overlap table, the corresponding survey area has no county record in the legend area overlap table and the corresponding survey area has no corresponding record in the survey area-county geographic coincidence table. Verify that this map unit is not included in the output.

For survey area XX002, only 3 map units have corresponding county records in the map unit area overlap table.

<b>Musym</b>	<b>Corresponding county or counties in map unit area overlap table</b>	<b>Expected number of output records</b>
001	NE001	1
002	NE003	1
003	NE001, NE003	2
004	None	3 (see explanation below)

Map unit 004, like all other map units for survey area XX002, has no corresponding county record in the map unit area overlap table. But survey area XX002 does have 3 county records in the legend area overlap table (NE001, NE003 and NC065). Therefore map unit 004 and all other map units in survey area XX002 that have no corresponding county record in the map unit area overlap table, should have 3 output records, one for NE001, one for NE003 and one for NC065.

Survey area XX003 has no records in the map unit area overlap table, and no records in the legend area overlap table. Survey area XX003 does have 2 records in the survey area-county geographic coincidence table. Therefore every map unit in survey area XX003 should have 2 output records, one for BR549 and one for AB123.

For my testing, I am using a survey area-county geographic coincidence table that contains only the following records:

STSSAID	STCOID
XX002	CO111
XX002	CO333
XX003	BR549
XX003	AB123

In the Access database, any map unit for which no corresponding county can be determined is logged to table “SYSTEM – FSA-CRP – Warnings and Rejects”.

### ***Component Selection and T Factor Testing***

Map units 101, 102 and 103 are used to test that the correct component is being selected as the “first” component. There isn’t any real logic to T Factor selection, so in these tests I’m just setting T Factor so that I know that the correct component was selected.

Musym	Test Scenario	Expected T factor
101	Derived RV comppect based on comppect_r, includes a number of components that should not be selected	1
102	Derived RV comppect based on (comppect_l + comppect_h)/2, includes a number of components that should not be selected	2
103	Derived RV comppect is Null, includes no components that should be selected	Null

In the Access database, any map unit for which no corresponding component can be determined IS output, but a warning is logged to table “SYSTEM – FSA-CRP – Warnings and Rejects”.

## ***Long Leaf Pine Suitability Testing***

Map units 201 through 221 are used to test the derivation of suitability for long leaf pine. Note that the only county for which a “Y” should ever occur is NC065, which occurs in the list of long leaf pine counties. For map units 201 through 221 in all other counties (NE001 and NE003), suitability for long leaf pine should be “N”, since those other counties are not in the list of long leaf pine counties.

<b>Musym</b>	<b>Test Scenario</b>	<b>Expected Suitability</b>
201	PIPA2 found in component existing plants	Y
202	PIPA2 found in component forest productivity	Y
203	PIPA2 found in component trees to manage	Y
204	All 3 aforementioned plant tables contain at least one plant, but never PIPA2	N
205	No explicit reference to PIPA2, fails wetness, fails texture, fails pH	N
206	No explicit reference to PIPA2, fails wetness, passes texture, passes pH	N
207	No explicit reference to PIPA2, passes wetness, fails texture, passes pH	N
208	No explicit reference to PIPA2, passes wetness, passes texture, fails pH (too high)	N
209	No explicit reference to PIPA2, passes wetness, passes texture, fails pH (null)	N
210-221	No explicit reference to PIPA2, passes wetness, passes texture, passes pH, every map unit in this set has a different valid texture	Y

### ***I Factor Testing***

Map units 301 through 310 are used to test the derivation of I Factor.

<b>Musym</b>	<b>WEI</b>	<b>WEG</b>	<b>Expected I Factor</b>
301	310	8	310
302	Null	1	220
303	Null	2	134
304	Null	3	86
305	Null	4	86
306	Null	4L	86
307	Null	5	56
308	Null	6	48
309	Null	7	38
310	Null	8	0

### ***K Factor Testing***

Map units 401 through 408 are used to test the derivation of K Factor.

<b>Musym</b>	<b>Test Scenario</b>	<b>Expected K Factor</b>
401	Test selection of maximum Kf for multiple layers at same depth	.10
402	Test selection of maximum Kw for multiple layers at same depth	.20
403	Test selection of Kf favored over Kw	.32
404	Test selection of Kw over a greater Kf in a lower layer	.02
405	Test selection of Kf over a greater Kw in a lower layer	.05
406	Test that no K Factor is returned for a Histosol	Null
407	Test that no K Factor is returned for a layer whose master designation is "O"	.24
408	Test that no K Factor is returned for a layer whose RV om is > 35%	.37

## ***LS Factor Testing***

Map units in the 500's are used to test the derivation of LS Factor. Survey area XX002 is not in the Palouse region, but survey area XX003 is in the Palouse region. Note that map units where slope is populated but slope length isn't, trigger a slope length lookup in the appropriate table, depending on whether or not the corresponding survey area is in the Palouse region.

<b>SSA</b>	<b>Musym</b>	<b>Slope</b>	<b>Slope length in meters</b>	<b>Expected LS Factor</b>
XX002	501		100	Null
XX002	502	0	0	.05
XX002	503	0	335	.06
XX002	504	61	0	.76
XX002	505	61	335	52.7
XX002	506	10	30	1.38
XX002	507	.5		.09
XX002	508	1		.16
XX002	509	2		.35
XX002	510	3		.48
XX002	511	4		.67
XX002	512	5		.76
XX002	513	6		.93
XX002	514	7		1.26
XX002	515	8		1.26
XX002	516	9		1.71
XX002	517	10		1.38
XX002	518	11		1.79
XX002	519	12		1.79
XX002	520	13		2.19
XX002	521	14		1.86
XX002	522	15		2.20
XX002	523	17		1.73
XX002	524	18		2.22
<b>XX003</b>	<b>525</b>	<b>2</b>		<b>.37</b>
<b>XX003</b>	<b>526</b>	<b>10</b>		<b>2.44</b>
<b>XX003</b>	<b>527</b>	<b>11</b>		<b>2.95</b>
<b>XX003</b>	<b>528</b>	<b>20</b>		<b>5.21</b>
<b>XX003</b>	<b>529</b>	<b>21</b>		<b>4.36</b>
<b>XX003</b>	<b>530</b>	<b>26</b>		<b>5.69</b>
<b>XX003</b>	<b>531</b>	<b>36</b>		<b>7.14</b>

### ***Undrained Leaching Index Testing***

Map units 601 through 619 are used to test the derivation of Undrained Leaching Index. Note that there are no specific tests to make sure that the correct Kw was returned. That's because the logic for selecting Kw is virtually identical to the logic for selecting K Factor, except in this case, only Kw is considered. This logic was already verified in the K Factor testing.

<b>Musym</b>	<b>Test Scenario</b>	<b>Expect Undrained Leaching Factor</b>
601	HG=C, calc<=10000, kval>=280	1
602	HG=C, calc>=10000	1
603	HG=B, calc>=35000, kval>=400	1
604	HG=B, calc>=45000, kval>=200	1
605	HG=A, calc<=30000	3
606	HG=B, calc<=9000, kval<=480	3
607	HG=B, calc<=15000, kval<=260	3
608	HG=A, calc>30000	2
609	HG=B and meets no other case where Hg=B	2
610	HG=C and meets no other case where Hg=C	2
611	Like 605 only HG=D	1
612	Like 606 only HG=A/D	1
613	Like 607 only HG=B/D	1
614	Like 608 only HG=C/D	1
615	Like 609 only taxorder=Histosol	1
616	Like 610 only HG is Null	Null
617	Like 601 only Kw is Null	Null
618	Like 602 only RV om is Null	Null
619	Like 603 only rv hzdepb is Null	Null

## The Gory Details

Below is the source code for the functions that generate the FSA-CRP data.

### ***Declarations***

```
Option Compare Database
Option Explicit
Global LSTable(1 To 19, 1 To 17) As Single
```

### ***CaseSensitiveMusym***

```
Function CaseSensitiveMusym(musym As String) As String
'
'This function expands a map unit symbol into a string that permits
'that map unit symbol to be treated as a case sensitive value.
'Each character in the original map unit symbol is expanded to 2
'characters. The first character is the original character that
'was encountered. The next character depends on the original character.
'
'A lower case letter is expanded to: lower case letter + "L".
'An upper case letter is expanded to: upper case letter + "U".
'A character that is not a lower case letter and is not an upper
'case letter is expanded to: character + "A".
'
'For example, map unit symbol "HoA21" expands to "HUoLAU2A1A".
'
Dim strMusym As Variant
Dim i As Long

strMusym = ""
```



```
For i = 1 To Len(musym)
    strMusym = strMusym & mid(musym, i, 1)
    If InStr(1, "ABCDEFGHJKLMNOPQRSTUVWXYZ", mid(musym, i, 1), 0) Then
        strMusym = strMusym & "U"
    ElseIf InStr(1, "abcdefghijklmnopqrstuvwxyz", mid(musym, i, 1), 0) Then
        strMusym = strMusym & "L"
    Else
        strMusym = strMusym & "A"
    End If
Next i

CaseSensitiveMusym = strMusym

End Function
```

### ***FSACRP\_Create\_Data***

```
Function FSACRP_Create_Data()
'
'This function derives a set of attributes that are used by
'the Farm Service Agency as part of the criteria as to whether
'or not a particular map unit is eligible for inclusion in the
'Conservation Reserve Program.
'
'This function always operates against ALL data that is currently
'loaded in the database.
'
Dim dbsSSURGO As Database

Dim qdfTemp As QueryDef
Dim strSQL As String

Dim rstFSACRP_Input As Recordset
Dim rstFSACRP_Output As Recordset
Dim rstFSACRP_Rejects As Recordset
```

```
Dim stcty As Variant
Dim ssaid As Variant
Dim musym As Variant
Dim mu_lleaf As Variant
Dim mu_leach As Variant
Dim mu_ifactor As Variant
Dim mu_kfactor As Variant
Dim mu_LS As Variant
Dim mu_tfactor As Variant
Dim source As Variant

Dim rstComponent As Recordset
Dim firstcokey As Variant
Dim hydgrp As Variant
Dim wei As Variant
Dim weg As Variant
Dim rvslope As Variant
Dim rvslopelenusle As Variant
Dim taxorder As Variant

Dim current_case_sensitive_musym As Variant
Dim current_area_symbol As Variant
Dim NoCompWarningMsg As String
Dim RejectMsg As String
Dim MapunitsWithNoCounty As Integer
Dim MapunitsWithNoCandidateComponent As Integer
Dim rstMUAOverlap As Recordset
Dim rstLAOverlap As Recordset
Dim rstSACoGeoCoincidence As Recordset
Dim rstLongLeaf As Recordset
Dim rstWetTopFoot As Recordset
Dim rstRequiredTexturesTopFoot As Recordset
Dim pHCriteriaMet As Boolean
Dim rstpHTopFoot As Recordset
Dim rstKFactorSurfaceMineralHorizon As Recordset
Dim rstULeachSurfaceMineralHorizon As Recordset
Dim hg As Variant
Dim calc As Variant
```

```
Dim kval As Variant
Dim blnInPalouse
Dim row As Integer
Dim column As Integer
Dim WarningMsg As String
Dim newline As String

newline = String(1, 13) & String(1, 10)

InitLSTable

RejectMsg = "Reject: This map unit WAS NOT included in the output file because no corresponding county
could be determined."
NoCompWarningMsg = "Warning: This map unit was included in the output file, but no values could be derived
because no candidate component could be determined."

Set dbsSSURGO = DBEngine.Workspaces(0).Databases(0)

'Drop any existing FSA-CRP data
Set qdfTemp = dbsSSURGO.CreateQueryDef("", "Delete from [SYSTEM - FSA-CRP Data]")
qdfTemp.Execute

'Drop any existing FSA-CRP rejects.
'Rejects are map units where no corresponding county could be determined.
Set qdfTemp = dbsSSURGO.CreateQueryDef("", "Delete from [SYSTEM - FSA-CRP - Warnings and Rejects]")
qdfTemp.Execute

Set rstFSACRP_Input = dbsSSURGO.OpenRecordset("FSA-CRP - Input", DB_OPEN_DYNASET)
Set rstFSACRP_Output = dbsSSURGO.OpenRecordset("SYSTEM - FSA-CRP Data", DB_OPEN_TABLE)
Set rstFSACRP_Rejects = dbsSSURGO.OpenRecordset("SYSTEM - FSA-CRP - Warnings and Rejects", DB_OPEN_TABLE)

current_area_symbol = ""
current_case_sensitive_musym = ""
MapunitsWithNoCounty = 0
MapunitsWithNoCandidateComponent = 0

Do Until rstFSACRP_Input.EOF
    'Only process of the first occurrence of a particular map unit symbol, based on sort on map unit status
```

```
'correlated > approved > provisional
If rstFSACRP_Input![Case Sensitive Musym] <> current_case_sensitive_musym Then
    current_case_sensitive_musym = rstFSACRP_Input![Case Sensitive Musym]

    If rstFSACRP_Input![areasympol] <> current_area_symbol Then
        'Area symbol has changed. Establish whether or not this survey area is in the Palouse region.
        current_area_symbol = rstFSACRP_Input![areasympol]
        blnInPalouse = InPalouse(rstFSACRP_Input![areasympol])
    End If

    'Establish defaults for all output values.
    stcty = Null
    ssaid = rstFSACRP_Input![areasympol]
    musym = rstFSACRP_Input![musym]
    mu_lleaf = "N"
    mu_leach = Null
    mu_ifactor = Null
    mu_kfactor = Null
    mu_LS = Null
    mu_tfactor = Null
    source = "S"

'*****
'** Select the "first" component for this map unit.
'** Establish the T Factor for this map unit at this time.
'** Save some other component level values that are used to derive other
'** attributes.
'*****

    firstcokey = Null
    strSQL = "SELECT RV([comppct_l],[comppct_r],[comppct_h]) AS [rv comppct], component.cokey, "
    strSQL = strSQL & "RV([slope_l],[slope_r],[slope_h]) AS [rv slope], "
    strSQL = strSQL & "RV([slopelenusle_l],[slopelenusle_r],[slopelenusle_h]) AS [rv slopelenusle], "
    strSQL = strSQL & "component.tfact, component.wei, component.weg, component.hydgrp, "
component.taxorder, component.mukey "
    strSQL = strSQL & "FROM component "
    strSQL = strSQL & "WHERE (((RV([comppct_l], [comppct_r], [comppct_h])) Is Not Null) And
((component.mukey) = '" & rstFSACRP_Input![mukey] & "') "

```

```
strSQL = strSQL & "ORDER BY CDb1(RV([comppct_l],[comppct_r],[comppct_h])) DESC , component.cokey;"
Set rstComponent = dbsSSURGO.OpenRecordset(strSQL)
If rstComponent.RecordCount <> 0 Then
    mu_tfactor = rstComponent![tfact]
    firstcokey = rstComponent![cokey]
    rvslope = rstComponent![rv slope]
    rvslopelenusle = rstComponent![rv slopelenusle]
    hydgrp = rstComponent![hydgrp]
    wei = rstComponent![wei]
    weg = rstComponent![weg]
    taxorder = rstComponent![taxorder]
Else
    'Log this map unit to the warnings and rejects table and increment the map units with no
candidate component warning count.
    rstFSACRP_Rejects.AddNew
    rstFSACRP_Rejects![areasympol] = rstFSACRP_Input![areasympol]
    rstFSACRP_Rejects![areaname] = rstFSACRP_Input![areaname]
    rstFSACRP_Rejects![musym] = rstFSACRP_Input![musym]
    rstFSACRP_Rejects![muname] = rstFSACRP_Input![muname]
    rstFSACRP_Rejects![mustatus] = rstFSACRP_Input![mustatus]
    rstFSACRP_Rejects![message] = NoCompWarningMsg
    rstFSACRP_Rejects.Update
    MapunitsWithNoCandidateComponent = MapunitsWithNoCandidateComponent + 1
End If
rstComponent.Close

'*****
'*** Establish the LS Factor for this map unit.
'*****

If Not IsNull(firstcokey) And Not IsNull(rvslope) Then
    If Not IsNull(rvslopelenusle) Then
        'Convert slope length in meters to feet.
        rvslopelenusle = rvslopelenusle * 3.28
    End If
    If IsNull(rvslopelenusle) Then
        'Establish slope length via table lookup.
        'In both lookup tables, slope length is in feet.
```

```
If Not blnInPalouse Then
    'Use Lightle and Weesies 1966 slope length lookup table.
    If rvslope >= 0 And rvslope < 0.75 Then
        rvslopelenusle = 100
    ElseIf rvslope >= 0.75 And rvslope < 1.5 Then
        rvslopelenusle = 200
    ElseIf rvslope >= 1.5 And rvslope < 2.5 Then
        rvslopelenusle = 300
    ElseIf rvslope >= 2.5 And rvslope < 3.5 Then
        rvslopelenusle = 200
    ElseIf rvslope >= 3.5 And rvslope < 4.5 Then
        rvslopelenusle = 180
    ElseIf rvslope >= 4.5 And rvslope < 5.5 Then
        rvslopelenusle = 160
    ElseIf rvslope >= 5.5 And rvslope < 6.5 Then
        rvslopelenusle = 150
    ElseIf rvslope >= 6.5 And rvslope < 7.5 Then
        rvslopelenusle = 140
    ElseIf rvslope >= 7.5 And rvslope < 8.5 Then
        rvslopelenusle = 130
    ElseIf rvslope >= 8.5 And rvslope < 9.5 Then
        rvslopelenusle = 125
    ElseIf rvslope >= 9.5 And rvslope < 10.5 Then
        rvslopelenusle = 120
    ElseIf rvslope >= 10.5 And rvslope < 11.5 Then
        rvslopelenusle = 110
    ElseIf rvslope >= 11.5 And rvslope < 12.5 Then
        rvslopelenusle = 100
    ElseIf rvslope >= 12.5 And rvslope < 13.5 Then
        rvslopelenusle = 90
    ElseIf rvslope >= 13.5 And rvslope < 14.5 Then
        rvslopelenusle = 80
    ElseIf rvslope >= 14.5 And rvslope < 15.5 Then
        rvslopelenusle = 70
    ElseIf rvslope >= 15.5 And rvslope < 17.5 Then
        rvslopelenusle = 60
    Else
        rvslopelenusle = 50
```

```
End If
Else
  'Use Palouse region slope length lookup table.
  If rvslope >= 0 And rvslope < 5.5 Then
    rvslopelenusle = 350
  ElseIf rvslope >= 5.5 And rvslope < 10.5 Then
    rvslopelenusle = 275
  ElseIf rvslope >= 10.5 And rvslope < 15.5 Then
    rvslopelenusle = 225
  ElseIf rvslope >= 15.5 And rvslope < 20.5 Then
    rvslopelenusle = 175
  ElseIf rvslope >= 20.5 And rvslope < 25.5 Then
    rvslopelenusle = 150
  ElseIf rvslope >= 25.5 And rvslope < 35.5 Then
    rvslopelenusle = 125
  Else
    rvslopelenusle = 100
  End If
End If
End If
'Establish row and column for LS Factor table lookup.
'Determine row index based on slope gradient.
If rvslope >= 0 And rvslope < 0.35 Then
  row = 1
ElseIf rvslope >= 0.35 And rvslope < 0.75 Then
  row = 2
ElseIf rvslope >= 0.75 And rvslope < 1.5 Then
  row = 3
ElseIf rvslope >= 1.5 And rvslope < 2.5 Then
  row = 4
ElseIf rvslope >= 2.5 And rvslope < 3.5 Then
  row = 5
ElseIf rvslope >= 3.5 And rvslope < 4.5 Then
  row = 6
ElseIf rvslope >= 4.5 And rvslope < 5.5 Then
  row = 7
ElseIf rvslope >= 5.5 And rvslope < 7 Then
  row = 8
```

```
ElseIf rvslope >= 7 And rvslope < 9 Then
    row = 9
ElseIf rvslope >= 9 And rvslope < 11 Then
    row = 10
ElseIf rvslope >= 11 And rvslope < 13 Then
    row = 11
ElseIf rvslope >= 13 And rvslope < 15 Then
    row = 12
ElseIf rvslope >= 15 And rvslope < 18 Then
    row = 13
ElseIf rvslope >= 18 And rvslope < 22.5 Then
    row = 14
ElseIf rvslope >= 22.5 And rvslope < 27.5 Then
    row = 15
ElseIf rvslope >= 27.5 And rvslope < 35 Then
    row = 16
ElseIf rvslope >= 35 And rvslope < 45 Then
    row = 17
ElseIf rvslope >= 45 And rvslope < 55 Then
    row = 18
Else
    row = 19
End If
'Determine column index based on slope length.
If rvslopelenusle >= 0 And rvslopelenusle < 4.5 Then
    column = 1
ElseIf rvslopelenusle >= 4.5 And rvslopelenusle < 7.5 Then
    column = 2
ElseIf rvslopelenusle >= 7.5 And rvslopelenusle < 10.5 Then
    column = 3
ElseIf rvslopelenusle >= 10.5 And rvslopelenusle < 13.5 Then
    column = 4
ElseIf rvslopelenusle >= 13.5 And rvslopelenusle < 20 Then
    column = 5
ElseIf rvslopelenusle >= 20 And rvslopelenusle < 37.5 Then
    column = 6
ElseIf rvslopelenusle >= 37.5 And rvslopelenusle < 62.5 Then
    column = 7
```



```
ElseIf rvslopelenusle >= 62.6 And rvslopelenusle < 87.5 Then
    column = 8
ElseIf rvslopelenusle >= 87.5 And rvslopelenusle < 125 Then
    column = 9
ElseIf rvslopelenusle >= 125 And rvslopelenusle < 175 Then
    column = 10
ElseIf rvslopelenusle >= 175 And rvslopelenusle < 225 Then
    column = 11
ElseIf rvslopelenusle >= 225 And rvslopelenusle < 275 Then
    column = 12
ElseIf rvslopelenusle >= 275 And rvslopelenusle < 350 Then
    column = 13
ElseIf rvslopelenusle >= 350 And rvslopelenusle < 500 Then
    column = 14
ElseIf rvslopelenusle >= 500 And rvslopelenusle < 700 Then
    column = 15
ElseIf rvslopelenusle >= 700 And rvslopelenusle < 900 Then
    column = 16
Else
    column = 17
End If
mu_LS = LSTable(row, column)
End If
```

```
'*****
'** Establish the I Factor for this map unit at this time.
'*****
```

```
If Not IsNull(firstcokey) Then
    If Not IsNull(wei) Then
        mu_ifactor = CInt(wei)
    ElseIf Not IsNull(weg) Then
        If weg = "1" Then
            mu_ifactor = 220
        ElseIf weg = "2" Then
            mu_ifactor = 134
        ElseIf weg = "3" Or weg = "4" Or weg = "4L" Then
            mu_ifactor = 86
```

```
        ElseIf weg = "5" Then
            mu_ifactor = 56
        ElseIf weg = "6" Then
            mu_ifactor = 48
        ElseIf weg = "7" Then
            mu_ifactor = 38
        ElseIf weg = "8" Then
            mu_ifactor = 0
        End If
    End If
End If

'*****
'** Establish suitability for long leaf pine for this map unit.
'*****

    If Not IsNull(firstcokey) Then
        'First look for plant symbol PIPA2 in the component trees to manage, component existing
vegetation
        'and component forest productivity tables.
        strSQL = "SELECT cotreestomng.plantsym, cotreestomng.cokey "
        strSQL = strSQL & "FROM cotreestomng "
        strSQL = strSQL & "WHERE (((cotreestomng.plantsym)='PIPA2') AND ((cotreestomng.cokey)='" &
firstcokey & "')) "
        strSQL = strSQL & "UNION "
        strSQL = strSQL & "SELECT coeplants.plantsym, coeplants.cokey "
        strSQL = strSQL & "FROM coeplants "
        strSQL = strSQL & "WHERE (((coeplants.plantsym)='PIPA2') AND ((coeplants.cokey)='" & firstcokey
& "')) "
        strSQL = strSQL & "UNION "
        strSQL = strSQL & "SELECT coforprod.plantsym, coforprod.cokey "
        strSQL = strSQL & "FROM coforprod "
        strSQL = strSQL & "WHERE (((coforprod.plantsym)='PIPA2') AND ((coforprod.cokey)='" & firstcokey
& "'))"; "

        Set rstLongLeaf = dbsSSURGO.OpenRecordset(strSQL)
        If rstLongLeaf.RecordCount <> 0 Then mu_lleaf = "Y"
        rstLongLeaf.Close
        If mu_lleaf = "N" Then
```

```
suitability      'No explicit reference to long leaf pine was found.  See if this component meets general
                  'requirements for long leaf pine.
the year.        'First determine if soil moisture status is "wet" in any layer < 30cm at any time during
                  strSQL = "SELECT RV([soilmoistdept_l],[soilmoistdept_r],[soilmoistdept_h]) AS [rv
soilmoistdept], cosoilmoist.soilmoiststat, comonth.cokey "
                  strSQL = strSQL & "FROM comonth INNER JOIN cosoilmoist ON comonth.comonthkey =
cosoilmoist.comonthkey "
                  strSQL = strSQL & "WHERE (((RV([soilmoistdept_l],[soilmoistdept_r],[soilmoistdept_h]))<30) AND
((cosoilmoist.soilmoiststat)='wet') AND ((comonth.cokey)='' & firstcokey & ''));"
                  Set rstWetTopFoot = dbsSSURGO.OpenRecordset(strSQL)
                  If rstWetTopFoot.RecordCount = 0 Then
                      'OK, there is no wetness problem in the top foot at any time of the year.
                      'Determine if any of the required textures ever occur in the top foot.
                      strSQL = "SELECT RV([hzdept_l],[hzdept_r],[hzdept_h]) AS [rv hzdept], chtexture.texcl,
chorizon.cokey "
                      strSQL = strSQL & "FROM (chorizon INNER JOIN chtexturegrp ON chorizon.chkey =
chttexturegrp.chkey) INNER JOIN chtexture ON chtexturegrp.chtgkey = chtexture.chtgkey "
                      strSQL = strSQL & "WHERE (((RV([hzdept_l],[hzdept_r],[hzdept_h]))<30) AND
((chttexture.texcl)='coarse sand' Or (chttexture.texcl)='sand' Or (chttexture.texcl)='fine sand' Or
(chttexture.texcl)='very fine sand' Or (chttexture.texcl)='loamy coarse sand' Or (chttexture.texcl)='loamy
sand' Or (chttexture.texcl)='loamy fine sand' Or (chttexture.texcl)='loamy very fine sand' Or
(chttexture.texcl)='coarse sandy loam' Or (chttexture.texcl)='sandy loam' Or (chttexture.texcl)='fine sandy
loam' Or (chttexture.texcl)='very fine sandy loam') AND ((chorizon.cokey)='' & firstcokey & ''));"
                      Set rstRequiredTexturesTopFoot = dbsSSURGO.OpenRecordset(strSQL)
                      If rstRequiredTexturesTopFoot.RecordCount <> 0 Then
                          'OK, at least one of the required textures was found in the top foot.
                          'Determine if the pH of every layer < 30cm is less than 6.0.
                          'Note that the existence of any layer < 30cm where pH cannot be determined
                          'results in a not suitable rating.
                          pHCriteriaMet = False
                          strSQL = "SELECT RV([hzdept_l],[hzdept_r],[hzdept_h]) AS [rv hzdept],
RV([ph1tolh2o_l],[ph1tolh2o_r],[ph1tolh2o_h]) AS [rv ph1tolh2o], chorizon.cokey "
                          strSQL = strSQL & "FROM chorizon "
                          strSQL = strSQL & "WHERE (((RV([hzdept_l],[hzdept_r],[hzdept_h]))<30) AND
((chorizon.cokey)='' & firstcokey & ''));"
                          Set rstpHTopFoot = dbsSSURGO.OpenRecordset(strSQL)
```

```
        If rstpHTopFoot.RecordCount <> 0 Then
            pHCriteriaMet = True
            Do Until rstpHTopFoot.EOF
                If IsNull(rstpHTopFoot![rv ph1tolh2o]) Or rstpHTopFoot![rv ph1tolh2o] >= 6
Then pHCriteriaMet = False
                    rstpHTopFoot.MoveNext
                Loop
            End If
            If pHCriteriaMet Then mu_lleaf = "Y"
            rstpHTopFoot.Close
        End If
        rstRequiredTexturesTopFoot.Close
    End If
    rstWetTopFoot.Close
End If
'
'Note than when data is output, suitability for long leaf pine is automatically reset to "N" if the
'corresponding county code does not occur in table "SYSTEM - Long Leaf Pine Counties".
'

'*****
'*** Establish the Undrained Leaching Factor for this map unit.
'*** The logic for determining Kw factor is virtually the same as the logic for
'*** determining the K Factor that is derived by this function, except that
'*** in this case, only Kw is considered.
'*****

        If Not IsNull(firstcokey) And taxorder = "Histosols" Then
            mu_leach = 1
        ElseIf Not IsNull(firstcokey) And Not IsNull(hydgrp) Then
            If hydgrp = "D" Or hydgrp = "A/D" Or hydgrp = "B/D" Or hydgrp = "C/D" Then
                mu_leach = 1
            Else
                strSQL = "SELECT RV([hzdept_l],[hzdept_r],[hzdept_h]) AS [rv hzdept],
RV([hzdepb_l],[hzdepb_r],[hzdepb_h]) AS [rv hzdepb], chorizon.kwfact, chorizon.desgnmaster,
RV([om_l],[om_r],[om_h]) AS [rv om], chorizon.cokey "
                strSQL = strSQL & "FROM chorizon "
```

```
strSQL = strSQL & "WHERE (((chorizon.kwfact) Is Not Null) " & _
    "And ((chorizon.desgnmaster) <> 'O' Or (chorizon.desgnmaster) Is Null)
" & _
    "And ((RV([om_l], [om_r], [om_h])) <= 35 Or (RV([om_l], [om_r],
[om_h])) Is Null) " & _
    "And ((chorizon.cokey) = ' " & firstcokey & "' ) " & _
    "And ((RV([hzdept_l], [hzdept_r], [hzdept_h])) Is Not Null)) "
strSQL = strSQL & "ORDER BY CDb1(RV([hzdept_l],[hzdept_r],[hzdept_h])), chorizon.kwfact
DESC;"

Set rstULeachSurfaceMineralHorizon = dbsSSURGO.OpenRecordset(strSQL)
If rstULeachSurfaceMineralHorizon.RecordCount <> 0 Then
    If Not IsNull(rstULeachSurfaceMineralHorizon![kwfact]) And Not
IsNull(rstULeachSurfaceMineralHorizon![rv om]) And Not IsNull(rstULeachSurfaceMineralHorizon![rv hzdepb])
Then
        hg = hydgrp
        calc = rstULeachSurfaceMineralHorizon![rv om] * (rstULeachSurfaceMineralHorizon![rv
hzdepb] / 2.54) * 1000
        kval = rstULeachSurfaceMineralHorizon![kwfact] * 1000
        If (hg = "D" Or hg = "A/D" Or hg = "B/D" Or hg = "C/D") Or _
            (hg = "C" And calc <= 10000 And kval >= 280) Or _
            (hg = "C" And calc >= 10000) Or _
            (hg = "B" And calc >= 35000 And kval >= 400) Or _
            (hg = "B" And calc >= 45000 And kval >= 200) Then
                mu_leach = 1
        ElseIf (hg = "A" And calc <= 30000) Or _
            (hg = "B" And calc <= 9000 And kval <= 480) Or _
            (hg = "B" And calc <= 15000 And kval <= 260) Then
                mu_leach = 3
        ElseIf (hg = "A" And calc > 30000) Or _
            (hg = "B" Or hg = "C") Then
                mu_leach = 2
        End If
    End If
End If
rstULeachSurfaceMineralHorizon.Close
End If
End If
```

```
*****
*** Establish the K Factor for this map unit.
*** The idea is to return the K Factor for the first mineral layer of a
*** non-histosol soil. If the soil is a histosol, no K Factor is returned.
*** This function is looking for the shallowest layer where either Kf or Kw
*** is not null, the horizon master designation is not "O", and RV organic
*** matter is <= 35%. If data is not populated correctly, this approach can
*** result in a K Factor not associated with the first mineral layer. We felt
*** that this was the best that we could do. Note that we do not exclude
*** layers from consideration when rv om or master designation cannot be
*** determined.
*****

    If Not IsNull(firstcokey) And (taxorder <> "Histosols" Or IsNull(taxorder)) Then
        strSQL = "SELECT RV([hzdept_l],[hzdept_r],[hzdept_h]) AS [rv hzdept], chorizon.kffact,
chorizon.kwfact, chorizon.desgnmaster, RV([om_l],[om_r],[om_h]) AS [rv om], chorizon.cokey "
        strSQL = strSQL & "FROM chorizon "
        strSQL = strSQL & "WHERE (((chorizon.kffact) Is Not Null) " & _
            "And ((chorizon.desgnmaster) <> 'O' Or (chorizon.desgnmaster) Is Null) " & _
            "And ((RV([om_l], [om_r], [om_h])) <= 35 Or (RV([om_l], [om_r], [om_h]))
Is Null) " & _
            "And ((chorizon.cokey) = '" & firstcokey & "') " & _
            "And ((RV([hzdept_l], [hzdept_r], [hzdept_h])) Is Not Null)) " & _
            "Or " & _
            "(((chorizon.kwfact) Is Not Null) " & _
            "And ((chorizon.desgnmaster) <> 'O' Or (chorizon.desgnmaster) Is Null) " & _
            "And ((RV([om_l], [om_r], [om_h])) <= 35 Or (RV([om_l], [om_r], [om_h]))
Is Null) " & _
            "And ((chorizon.cokey) = '" & firstcokey & "') " & _
            "And ((RV([hzdept_l], [hzdept_r], [hzdept_h])) Is Not Null)) "
        strSQL = strSQL & "ORDER BY CDb1(RV([hzdept_l],[hzdept_r],[hzdept_h])), chorizon.kffact DESC ,
chorizon.kwfact DESC;"
        Set rstKFactorSurfaceMineralHorizon = dbsSSURGO.OpenRecordset(strSQL)
        If rstKFactorSurfaceMineralHorizon.RecordCount <> 0 Then
            If Not IsNull(rstKFactorSurfaceMineralHorizon![kffact]) Then
                mu_kfactor = CSng(rstKFactorSurfaceMineralHorizon![kffact])
            End If
        End If
    End If
```

```
        ElseIf Not IsNull(rstKFactorSurfaceMineralHorizon![kwfact]) Then
            mu_kfactor = CSng(rstKFactorSurfaceMineralHorizon![kwfact])
        End If
    End If
    rstKFactorSurfaceMineralHorizon.Close
End If

'*****
'**  Output one record for this map unit for each corresponding county or parish,
'**  if any.
'*****

'Figure out if we can determine the corresponding county or counties.
'First look for corresponding county records in the map unit area overlap table.
'If we can't find any county overlaps for this map unit in the map unit area overlap table,
'see if this survey has any corresponding county records in the legend area overlap table.
'If we can't find any county overlaps for this survey in the legend area overlap table,
'see if this survey has any corresponding records in the survey area-county coincidence table.
'If we cant find any county overlaps for this survey area in the survey area-county coincidence
'table, log the problem and increment the map units with no county error count.

'Check for county coincidences in the map unit area overlap table.
strSQL = "SELECT laoverlap.areatypename, laoverlap.areasymbol, muaoverlap.mukey "
strSQL = strSQL & "FROM laoverlap INNER JOIN muaoverlap ON laoverlap.lareaovkey = "
muaoverlap.lareaovkey "
strSQL = strSQL & "WHERE (((laoverlap.areatypename)='county or parish') AND ((muaoverlap.mukey)='"
& rstFSACRP_Input![mukey] & "')));"
Set rstMUAOverlap = dbsSSURGO.OpenRecordset(strSQL)
If rstMUAOverlap.RecordCount <> 0 Then
    'Output a record for this map unit for each corresponding county or parish record in the map
    unit area overlap table.
    Do Until rstMUAOverlap.EOF
        rstFSACRP_Output.AddNew
        rstFSACRP_Output![stcty] = rstMUAOverlap![areasymbol]
        rstFSACRP_Output![ssaid] = ssaid
        rstFSACRP_Output![musym] = musym
        rstFSACRP_Output![mu_lleaf] = IIf(IsLongLeafPineCounty(rstMUAOverlap![areasymbol]),
mu_lleaf, "N")
    Loop
End If
```

```
rstFSACRP_Output![mu_leach] = mu_leach
rstFSACRP_Output![mu_ifactor] = mu_ifactor
rstFSACRP_Output![mu_kfactor] = mu_kfactor
rstFSACRP_Output![mu_LS] = mu_LS
rstFSACRP_Output![mu_tfactor] = mu_tfactor
rstFSACRP_Output![source] = source
rstFSACRP_Output.Update
rstMUAOverlap.MoveNext
Loop
rstMUAOverlap.Close
Else
rstMUAOverlap.Close
'Check for county coincidences for this SSA in the legend area overlap table.
strSQL = "SELECT laoverlap.areatypename, laoverlap.areasymbol, laoverlap.lkey "
strSQL = strSQL & "FROM laoverlap "
strSQL = strSQL & "WHERE (((laoverlap.areatypename)='County or Parish') AND
((laoverlap.lkey)='" & rstFSACRP_Input![lkey] & "')));"
Set rstLAOverlap = dbsSSURGO.OpenRecordset(strSQL)
If rstLAOverlap.RecordCount <> 0 Then
'Output a record for this map unit for each corresponding county or parish record in the
legend area overlap table.
Do Until rstLAOverlap.EOF
rstFSACRP_Output.AddNew
rstFSACRP_Output![stcty] = rstLAOverlap![areasymbol]
rstFSACRP_Output![ssaid] = ssaid
rstFSACRP_Output![musym] = musym
rstFSACRP_Output![mu_lleaf] = IIf(IsLongLeafPineCounty(rstLAOverlap![areasymbol]),
mu_lleaf, "N")

rstFSACRP_Output![mu_leach] = mu_leach
rstFSACRP_Output![mu_ifactor] = mu_ifactor
rstFSACRP_Output![mu_kfactor] = mu_kfactor
rstFSACRP_Output![mu_LS] = mu_LS
rstFSACRP_Output![mu_tfactor] = mu_tfactor
rstFSACRP_Output![source] = source
rstFSACRP_Output.Update
rstLAOverlap.MoveNext
Loop
rstLAOverlap.Close
```



```
Else
    rstLAOverlap.Close
    'Check for county coincidences for this SSA in the survey area-county geographic
coincidence table.
    '*****
    'What if the access database contains more than one instance of the same SSA?
    '*****
    strSQL = "SELECT [SYSTEM - Survey Area-County Geographic Coincidence].stssaid, [SYSTEM -
Survey Area-County Geographic Coincidence].stcoid "
    strSQL = strSQL & "FROM [SYSTEM - Survey Area-County Geographic Coincidence] "
    strSQL = strSQL & "WHERE ((([SYSTEM - Survey Area-County Geographic
Coincidence].stssaid)='" & rstFSACRP_Input![areasympol] & "')));"
    Set rstSACoGeoCoincidence = dbsSSURGO.OpenRecordset(strSQL)
    If rstSACoGeoCoincidence.RecordCount <> 0 Then
        'Output a record for this map unit for each corresponding county or parish in the
survey araa-county geogrphic coincidence table.
        Do Until rstSACoGeoCoincidence.EOF
            rstFSACRP_Output.AddNew
            rstFSACRP_Output![stcty] = rstSACoGeoCoincidence![stcoid]
            rstFSACRP_Output![ssaid] = ssaid
            rstFSACRP_Output![musym] = musym
            rstFSACRP_Output![mu_lleaf] =
IIf(IsLongLeafPineCounty(rstSACoGeoCoincidence![stcoid]), mu_lleaf, "N")
            rstFSACRP_Output![mu_leach] = mu_leach
            rstFSACRP_Output![mu_ifactor] = mu_ifactor
            rstFSACRP_Output![mu_kfactor] = mu_kfactor
            rstFSACRP_Output![mu_LS] = mu_LS
            rstFSACRP_Output![mu_tfactor] = mu_tfactor
            rstFSACRP_Output![source] = source
            rstFSACRP_Output.Update
            rstSACoGeoCoincidence.MoveNext
        Loop
        rstSACoGeoCoincidence.Close
    Else
        rstSACoGeoCoincidence.Close
        'Log this map unit to the warnings and rejects table and increment the map units with
no county error count.
        rstFSACRP_Rejects.AddNew
```

```
        rstFSACRP_Rejects![areasympbol] = rstFSACRP_Input![areasympbol]
        rstFSACRP_Rejects![areaname] = rstFSACRP_Input![areaname]
        rstFSACRP_Rejects![musym] = rstFSACRP_Input![musym]
        rstFSACRP_Rejects![muname] = rstFSACRP_Input![muname]
        rstFSACRP_Rejects![mustatus] = rstFSACRP_Input![mustatus]
        rstFSACRP_Rejects![message] = RejectMsg
        rstFSACRP_Rejects.Update
        MapunitsWithNoCounty = MapunitsWithNoCounty + 1
    End If
End If
End If
'Fetch the next input map unit record.
rstFSACRP_Input.MoveNext
Loop

rstFSACRP_Input.Close
rstFSACRP_Output.Close
rstFSACRP_Rejects.Close

If MapunitsWithNoCounty <> 0 Or MapunitsWithNoCandidateComponent <> 0 Then
    WarningMsg = ""
    If MapunitsWithNoCounty Then
        WarningMsg = "Rejects: Number of map units not output because no corresponding county or parish
could be determined: " & MapunitsWithNoCounty & newline & newline
    End If
    If MapunitsWithNoCandidateComponent Then
        WarningMsg = WarningMsg & "Warnings: Number of map units for which values could not be derived
because no candidate component could be determined: " & MapunitsWithNoCandidateComponent & newline &
newline
    End If
    WarningMsg = WarningMsg & "See table " & "SYSTEM - FSA-CRP - Warnings and Rejects" & " for the complete list
of map units for which valid data could not be derived."
    MsgBox WarningMsg, vbOKOnly + vbExclamation, "FSA-CRP Data Generation - Warnings"
End If

FSACRP_Create_Data = 0
```

End Function

### ***InitLSTable***

```
Function InitLSTable()  
,  
'This function initializes table 4-2 from Ag. Handbook  
'703. This table is used to derive topographical factor, LS,  
'for moderate ratio of rill to interrill erosion. The lookup  
'is based on slope gradient and USLE slope length.  
,  
'Dim i, j As Integer  
'Dim record As String  
,  
LSTable(1, 1) = 0.05  
LSTable(1, 2) = 0.05  
LSTable(1, 3) = 0.05  
LSTable(1, 4) = 0.05  
LSTable(1, 5) = 0.05  
LSTable(1, 6) = 0.05  
LSTable(1, 7) = 0.05  
LSTable(1, 8) = 0.05  
LSTable(1, 9) = 0.05  
LSTable(1, 10) = 0.05  
LSTable(1, 11) = 0.05  
LSTable(1, 12) = 0.05  
LSTable(1, 13) = 0.05  
LSTable(1, 14) = 0.05  
LSTable(1, 15) = 0.06  
LSTable(1, 16) = 0.06  
LSTable(1, 17) = 0.06  
LSTable(2, 1) = 0.07  
LSTable(2, 2) = 0.07  
LSTable(2, 3) = 0.07  
LSTable(2, 4) = 0.07  
LSTable(2, 5) = 0.07
```

LSTable(2, 6) = 0.08  
LSTable(2, 7) = 0.08  
LSTable(2, 8) = 0.08  
LSTable(2, 9) = 0.09  
LSTable(2, 10) = 0.09  
LSTable(2, 11) = 0.09  
LSTable(2, 12) = 0.09  
LSTable(2, 13) = 0.09  
LSTable(2, 14) = 0.1  
LSTable(2, 15) = 0.1  
LSTable(2, 16) = 0.1  
LSTable(2, 17) = 0.1  
LSTable(3, 1) = 0.11  
LSTable(3, 2) = 0.11  
LSTable(3, 3) = 0.11  
LSTable(3, 4) = 0.11  
LSTable(3, 5) = 0.11  
LSTable(3, 6) = 0.12  
LSTable(3, 7) = 0.13  
LSTable(3, 8) = 0.14  
LSTable(3, 9) = 0.14  
LSTable(3, 10) = 0.15  
LSTable(3, 11) = 0.16  
LSTable(3, 12) = 0.17  
LSTable(3, 13) = 0.17  
LSTable(3, 14) = 0.18  
LSTable(3, 15) = 0.19  
LSTable(3, 16) = 0.2  
LSTable(3, 17) = 0.2  
LSTable(4, 1) = 0.17  
LSTable(4, 2) = 0.17  
LSTable(4, 3) = 0.17  
LSTable(4, 4) = 0.17  
LSTable(4, 5) = 0.17  
LSTable(4, 6) = 0.19  
LSTable(4, 7) = 0.22  
LSTable(4, 8) = 0.25  
LSTable(4, 9) = 0.27

LSTable(4, 10) = 0.29  
LSTable(4, 11) = 0.31  
LSTable(4, 12) = 0.33  
LSTable(4, 13) = 0.35  
LSTable(4, 14) = 0.37  
LSTable(4, 15) = 0.41  
LSTable(4, 16) = 0.44  
LSTable(4, 17) = 0.47  
LSTable(5, 1) = 0.22  
LSTable(5, 2) = 0.22  
LSTable(5, 3) = 0.22  
LSTable(5, 4) = 0.22  
LSTable(5, 5) = 0.22  
LSTable(5, 6) = 0.25  
LSTable(5, 7) = 0.32  
LSTable(5, 8) = 0.36  
LSTable(5, 9) = 0.39  
LSTable(5, 10) = 0.44  
LSTable(5, 11) = 0.48  
LSTable(5, 12) = 0.52  
LSTable(5, 13) = 0.55  
LSTable(5, 14) = 0.6  
LSTable(5, 15) = 0.68  
LSTable(5, 16) = 0.75  
LSTable(5, 17) = 0.8  
LSTable(6, 1) = 0.26  
LSTable(6, 2) = 0.26  
LSTable(6, 3) = 0.26  
LSTable(6, 4) = 0.26  
LSTable(6, 5) = 0.26  
LSTable(6, 6) = 0.31  
LSTable(6, 7) = 0.4  
LSTable(6, 8) = 0.47  
LSTable(6, 9) = 0.52  
LSTable(6, 10) = 0.6  
LSTable(6, 11) = 0.67  
LSTable(6, 12) = 0.72  
LSTable(6, 13) = 0.77

LSTable(6, 14) = 0.86  
LSTable(6, 15) = 0.99  
LSTable(6, 16) = 1.1  
LSTable(6, 17) = 1.19  
LSTable(7, 1) = 0.3  
LSTable(7, 2) = 0.3  
LSTable(7, 3) = 0.3  
LSTable(7, 4) = 0.3  
LSTable(7, 5) = 0.3  
LSTable(7, 6) = 0.37  
LSTable(7, 7) = 0.49  
LSTable(7, 8) = 0.58  
LSTable(7, 9) = 0.65  
LSTable(7, 10) = 0.76  
LSTable(7, 11) = 0.85  
LSTable(7, 12) = 0.93  
LSTable(7, 13) = 1.01  
LSTable(7, 14) = 1.13  
LSTable(7, 15) = 1.33  
LSTable(7, 16) = 1.49  
LSTable(7, 17) = 1.63  
LSTable(8, 1) = 0.34  
LSTable(8, 2) = 0.34  
LSTable(8, 3) = 0.34  
LSTable(8, 4) = 0.34  
LSTable(8, 5) = 0.34  
LSTable(8, 6) = 0.43  
LSTable(8, 7) = 0.58  
LSTable(8, 8) = 0.69  
LSTable(8, 9) = 0.78  
LSTable(8, 10) = 0.93  
LSTable(8, 11) = 1.05  
LSTable(8, 12) = 1.16  
LSTable(8, 13) = 1.25  
LSTable(8, 14) = 1.42  
LSTable(8, 15) = 1.69  
LSTable(8, 16) = 1.91  
LSTable(8, 17) = 2.11

LSTable(9, 1) = 0.42  
LSTable(9, 2) = 0.42  
LSTable(9, 3) = 0.42  
LSTable(9, 4) = 0.42  
LSTable(9, 5) = 0.42  
LSTable(9, 6) = 0.53  
LSTable(9, 7) = 0.74  
LSTable(9, 8) = 0.91  
LSTable(9, 9) = 1.04  
LSTable(9, 10) = 1.26  
LSTable(9, 11) = 1.45  
LSTable(9, 12) = 1.62  
LSTable(9, 13) = 1.77  
LSTable(9, 14) = 2.03  
LSTable(9, 15) = 2.47  
LSTable(9, 16) = 2.83  
LSTable(9, 17) = 3.15  
LSTable(10, 1) = 0.46  
LSTable(10, 2) = 0.48  
LSTable(10, 3) = 0.5  
LSTable(10, 4) = 0.51  
LSTable(10, 5) = 0.52  
LSTable(10, 6) = 0.67  
LSTable(10, 7) = 0.97  
LSTable(10, 8) = 1.19  
LSTable(10, 9) = 1.38  
LSTable(10, 10) = 1.71  
LSTable(10, 11) = 1.98  
LSTable(10, 12) = 2.22  
LSTable(10, 13) = 2.44  
LSTable(10, 14) = 2.84  
LSTable(10, 15) = 3.5  
LSTable(10, 16) = 4.06  
LSTable(10, 17) = 4.56  
LSTable(11, 1) = 0.47  
LSTable(11, 2) = 0.53  
LSTable(11, 3) = 0.58  
LSTable(11, 4) = 0.61

LSTable(11, 5) = 0.64  
LSTable(11, 6) = 0.84  
LSTable(11, 7) = 1.23  
LSTable(11, 8) = 1.53  
LSTable(11, 9) = 1.79  
LSTable(11, 10) = 2.23  
LSTable(11, 11) = 2.61  
LSTable(11, 12) = 2.95  
LSTable(11, 13) = 3.26  
LSTable(11, 14) = 3.81  
LSTable(11, 15) = 4.75  
LSTable(11, 16) = 5.56  
LSTable(11, 17) = 6.28  
LSTable(12, 1) = 0.48  
LSTable(12, 2) = 0.58  
LSTable(12, 3) = 0.65  
LSTable(12, 4) = 0.7  
LSTable(12, 5) = 0.75  
LSTable(12, 6) = 1  
LSTable(12, 7) = 1.48  
LSTable(12, 8) = 1.86  
LSTable(12, 9) = 2.19  
LSTable(12, 10) = 2.76  
LSTable(12, 11) = 3.25  
LSTable(12, 12) = 3.69  
LSTable(12, 13) = 4.09  
LSTable(12, 14) = 4.82  
LSTable(12, 15) = 6.07  
LSTable(12, 16) = 7.15  
LSTable(12, 17) = 8.11  
LSTable(13, 1) = 0.49  
LSTable(13, 2) = 0.63  
LSTable(13, 3) = 0.72  
LSTable(13, 4) = 0.79  
LSTable(13, 5) = 0.85  
LSTable(13, 6) = 1.15  
LSTable(13, 7) = 1.73  
LSTable(13, 8) = 2.2



LSTable(13, 9) = 2.6  
LSTable(13, 10) = 3.3  
LSTable(13, 11) = 3.9  
LSTable(13, 12) = 4.45  
LSTable(13, 13) = 4.95  
LSTable(13, 14) = 5.86  
LSTable(13, 15) = 7.43  
LSTable(13, 16) = 8.79  
LSTable(13, 17) = 10.02  
LSTable(14, 1) = 0.52  
LSTable(14, 2) = 0.71  
LSTable(14, 3) = 0.85  
LSTable(14, 4) = 0.96  
LSTable(14, 5) = 1.06  
LSTable(14, 6) = 1.45  
LSTable(14, 7) = 2.22  
LSTable(14, 8) = 2.85  
LSTable(14, 9) = 3.4  
LSTable(14, 10) = 4.36  
LSTable(14, 11) = 5.21  
LSTable(14, 12) = 5.97  
LSTable(14, 13) = 6.68  
LSTable(14, 14) = 7.97  
LSTable(14, 15) = 10.23  
LSTable(14, 16) = 12.2  
LSTable(14, 17) = 13.99  
LSTable(15, 1) = 0.56  
LSTable(15, 2) = 0.8  
LSTable(15, 3) = 1  
LSTable(15, 4) = 1.16  
LSTable(15, 5) = 1.3  
LSTable(15, 6) = 1.81  
LSTable(15, 7) = 2.82  
LSTable(15, 8) = 3.65  
LSTable(15, 9) = 4.39  
LSTable(15, 10) = 5.69  
LSTable(15, 11) = 6.83  
LSTable(15, 12) = 7.88

LSTable(15, 13) = 8.86  
LSTable(15, 14) = 10.65  
LSTable(15, 15) = 13.8  
LSTable(15, 16) = 16.58  
LSTable(15, 17) = 19.13  
LSTable(16, 1) = 0.59  
LSTable(16, 2) = 0.89  
LSTable(16, 3) = 1.13  
LSTable(16, 4) = 1.34  
LSTable(16, 5) = 1.53  
LSTable(16, 6) = 2.15  
LSTable(16, 7) = 3.39  
LSTable(16, 8) = 4.42  
LSTable(16, 9) = 5.34  
LSTable(16, 10) = 6.98  
LSTable(16, 11) = 8.43  
LSTable(16, 12) = 9.76  
LSTable(16, 13) = 11.01  
LSTable(16, 14) = 13.3  
LSTable(16, 15) = 17.37  
LSTable(16, 16) = 20.99  
LSTable(16, 17) = 24.31  
LSTable(17, 1) = 0.65  
LSTable(17, 2) = 1.05  
LSTable(17, 3) = 1.38  
LSTable(17, 4) = 1.68  
LSTable(17, 5) = 1.95  
LSTable(17, 6) = 2.77  
LSTable(17, 7) = 4.45  
LSTable(17, 8) = 5.87  
LSTable(17, 9) = 7.14  
LSTable(17, 10) = 9.43  
LSTable(17, 11) = 11.47  
LSTable(17, 12) = 13.37  
LSTable(17, 13) = 15.14  
LSTable(17, 14) = 18.43  
LSTable(17, 15) = 24.32  
LSTable(17, 16) = 29.6

```
LSTable(17, 17) = 34.48
LSTable(18, 1) = 0.71
LSTable(18, 2) = 1.18
LSTable(18, 3) = 1.59
LSTable(18, 4) = 1.97
LSTable(18, 5) = 2.32
LSTable(18, 6) = 3.32
LSTable(18, 7) = 5.4
LSTable(18, 8) = 7.17
LSTable(18, 9) = 8.78
LSTable(18, 10) = 11.66
LSTable(18, 11) = 14.26
LSTable(18, 12) = 16.67
LSTable(18, 13) = 18.94
LSTable(18, 14) = 23.17
LSTable(18, 15) = 30.78
LSTable(18, 16) = 37.65
LSTable(18, 17) = 44.02
LSTable(19, 1) = 0.76
LSTable(19, 2) = 1.3
LSTable(19, 3) = 1.78
LSTable(19, 4) = 2.23
LSTable(19, 5) = 2.65
LSTable(19, 6) = 3.81
LSTable(19, 7) = 6.24
LSTable(19, 8) = 8.33
LSTable(19, 9) = 10.23
LSTable(19, 10) = 13.65
LSTable(19, 11) = 16.76
LSTable(19, 12) = 19.64
LSTable(19, 13) = 22.36
LSTable(19, 14) = 27.45
LSTable(19, 15) = 36.63
LSTable(19, 16) = 44.96
LSTable(19, 17) = 52.7
'
```

```
'The following code was used to create a hard copy table that
'I could use to check against the original copy that was provided.
```

```
'
'Open "d:\tmp5\table.txt" For Output As #1
'record = ""
'For i = 1 To 19
'    For j = 1 To 17
'        record = record & Format(LSTable(i, j), "00.00") & "  "
'    Next j
'    Print #1, record
'    record = ""
'Next i
'Close #1

End Function
```

### ***InPalouse***

```
Function InPalouse(areasybol As Variant) As Boolean
'
'This function determines if the area symbol that is passed to
'this function is a member of the set of soil survey areas contained
'in table "SYSTEM - Palouse Soil Survey Areas".
'
'This function returns either True or False.
'
Dim dbsSSURGO As Database
Dim qdfTemp As QueryDef
Dim strSQL As String
Dim rstPalouse As Recordset

InPalouse = False

If Not IsNull(areasybol) Then
    strSQL = "SELECT [SYSTEM - Palouse Soil Survey Areas].STSSAID "
    strSQL = strSQL & "FROM [SYSTEM - Palouse Soil Survey Areas] "
    strSQL = strSQL & "WHERE ((([SYSTEM - Palouse Soil Survey Areas].STSSAID)='" & areasybol & "')));"
    Set dbsSSURGO = DBEngine.Workspaces(0).Databases(0)
```

```
Set rstPalouse = dbsSSURGO.OpenRecordset(strSQL)
If rstPalouse.RecordCount <> 0 Then InPalouse = True
rstPalouse.Close
End If

End Function
```

### ***IsLongLeafPineCounty***

```
Function IsLongLeafPineCounty(varCountyCode As Variant) As Boolean
'
'This function determines if the argument passed in occurs in column
' "countycode" of table "SYSTEM - Long Leaf Pine Counties".
'
Dim dbsSSURGO As Database
Dim strSQL As String
Dim rstLongLeafPine As Recordset

IsLongLeafPineCounty = False
If Not IsNull(varCountyCode) And Len(varCountyCode) <> 0 Then
    strSQL = "SELECT [SYSTEM - Long Leaf Pine Counties].countycode "
    strSQL = strSQL & "FROM [SYSTEM - Long Leaf Pine Counties] "
    strSQL = strSQL & "WHERE ((([SYSTEM - Long Leaf Pine Counties].countycode)='" & varCountyCode & "'));"
    Set dbsSSURGO = CurrentDb()
    Set rstLongLeafPine = dbsSSURGO.OpenRecordset(strSQL)
    If rstLongLeafPine.RecordCount <> 0 Then IsLongLeafPineCounty = True
    rstLongLeafPine.Close
End If
End Function
```

### ***RV***

```
Function RV(value_l As Variant, value_r As Variant, value_h As Variant) As Variant
'
```

```
'This function returns a derived representative value for a low, rv, high attribute.
',
'If the rv value is not null, the rv value is returned.
'If the rv value is null, but the low and high values are not null,
'(low value + high value)/2 is returned.
'If the rv value is null, and either the low value is null, or the
'high value is null, Null is returned.
',
If Not IsNull(value_r) Then
    RV = value_r
ElseIf Not IsNull(value_l) And Not IsNull(value_h) Then
    RV = (value_l + value_h) / 2
Else
    RV = Null
End If
End Function
```